

Artículo/Article
Sección C/Section C
Vol. 15, nro. 2
e2932

Solución a un problema de eventos discretos en sistemas dinámicos utilizando las redes de Petri y el software HiPS

Saul Manuel Favela Camacho[1*]

[1]Universidad Popular Autónoma del Estado de Puebla, Programa del Doctorado en Ingeniería en Mecatrónica; 21 sur #1103, Barrio de Santiago, Puebla, Puebla, 72410, México.

*Autor para correspondencia/Corresponding author: saulmanuel.favela@upaep.edu.mx

Discrete event problem solution in dynamic systems using Petri networks and HiPS software

Resumen

En este trabajo se exponen los pasos y la solución del problema del gato y el ratón que se publicó en el artículo de Wonham y Ramadge (1984), utilizando el modelo de las redes de Petri y el software de simulación *Hierarchical Petri Net Simulator* (HiPS). El objetivo que se persigue es mostrar la familiaridad y la metodología de la aplicación de lenguage controlable en eventos discretos utilizando las herramientas señaladas, puesto que el artículo fue escrito hace casi 39 años y no se tenía ventaja de los sistemas informáticos existentes en los que se pueden simular eventos discretos. Además de mostrar el esquema de control de redes de Petri que permite la solución del problema bajo el cumplimiento de ciertas condiciones, se toma ventaja del uso del software de simulación, donde se permite realizar una tabla de alcanzabilidad y covertura del modelo, una matriz de incidencia, la identificación de estados que no tienen conflicto y transiciones para ejectuar esos estados sin restricciones. En términos simbólicos, el comportamiento del gato y el ratón pueden representar sensores, actuadores, equipos, robots, maquinaria o sistemas de producción dinámicos. Los resultados fueron óptimos y se discuten conclusiones.

Palabras clave: redes de Petri, eventos discretos, teoría de control, sistemas dinámicos, lenguaje controlable

Abstract

In this work are exposed the solution steps of the cat and mouse problem published in the Wonham y Ramadge's article from 1984, using Petri networks models and the *Hiearchical Petri Net Simulator Software* (HiPS). The objective is to show familiarity and the methodology of applying controllable language in discrete events using the previous tools, since the article was written almost 39 years ago without the advantages of information technologies where discrete events can be simulated. Besides showing the control schematics of Petri networks that allows the problem solution under fulfillment of certain conditions, the simulation software is used in advantage to get the model's coverage and reachability table, an incidence matrix, the identification of out-of-conflict states and transitions to





Editado por / Edited by: Dennis Cazar

Recibido / Received: 31/03/2023

Aceptado / Accepted: 07/06/2023

Publicado en línea / Published online: 18/10/2023

 DOI: https://doi.org/10.18272/aci.v15i2.2932





execute those states without restrictions. In symbolic terms, the behavior of the cat and the mouse can represent equipment, sensors, actuators, robots, machinery, or dynamic production systems. The results were optimal, and conclusions are discussed.

Keywords: Petri Networks, discrete events, control theory, dynamic systems, controllable language

INTRODUCCIÓN

La temática de la teoría de control y su implementación práctica son áreas importantes en la aplicación y soluciones en diferentes campos de la actividad humana [1]. En esa temática se incluyen algunos métodos de control para los sistemas de variable discreta, donde los cambios en los mismos se manifiestan como eventos que se "activan" o "disparan", ya sea por una acción externa o un cambio en alguno o en todos los componentes que la conforman. En los ambientes de manufactura donde hay controles, redes de comunicación e intercambio de información con otros sistemas y subsistemas, manifestándose los eventos discretos [2,3], se puede utilizar un modelo de representación gráfica y matemática para simular el comportamiento discreto. El modelo es conocido como las redes de Petri.

Creados por el Dr. Carl Adam Petri en 1962, son una herramienta de modelación que puede ser aplicable a cualquier sistema que se caracterice por ser concurrente, asíncrono, distribuido, paralelo, no determinístico y/o estocástico. Como herramienta gráfica, las redes de Petri se utilizan para representar la interacción entre los elementos o subsistemas, similar a un diagrama de flujo. Y como herramienta matemática, se pueden utilizar ecuaciones de estado, ecuaciones algebraicas y otros modelos matemáticos que simulen el comportamiento del sistema [4]. Esta combinación de visualización gráfica y modelación matemática ha hecho popular su utilización para representar sistemas dinámicos, como procesos de manufactura, sistemas de control y redes logísticas [5].

Una red de Petri de manera general es una representación gráfica que está conformada por diversos elementos: los círculos que representan las plazas; los rectángulos a las transiciones; los puntos dentro de los círculos simbolizan los marcadores; y los arcos señalan las relaciones o conexiones entre las plazas y las transiciones [6,7]. Con relación a las plazas, estas pueden ser precondiciones, condiciones o postcondiciones, información de entrada o de salida, señales de entrada o de salida, recursos necesarios y liberados, búferes y condiciones; en cambio, las transiciones pueden ser eventos, pasos computacionales, procesamiento de señales, una tarea o acción y cláusulas; y los marcadores representan estados [4].

Sin embargo, los trabajos hechos por Ramadge y Wonham [8-12] se consideran como los pioneros en el campo de los sistemas de eventos discretos. Ellos propusieron modelos en donde los eventos, están divididos en dos clases: eventos controlables e incontrolables, donde en los primeros existen condiciones del sistema que pueden ser activados o desactivados a voluntad para influir en el comportamiento del mismo.

 DOI: https://doi.org/10.18272/aci.v15i2.2932





Mediante la interacción que se realiza con esos eventos, se van reproduciendo resultados. Por tal motivo, se requiere de sistemas de control para que esos eventos se lleven a cabo conforme al comportamiento deseado, y que resulte también en resultados esperados. La solución de algunos problemas de control surge desde la perspectiva que presentaron los autores hace más de tres décadas para entender la dinámica de los sistemas de control de eventos discretos. Sin embargo, a medida que ha evolucionado la tecnología, se tiene a la disposición algunas herramientas informáticas que permiten simular eventos discretos y problemas de control para resolverlos de forma más rápida y eficiente.

El siguiente trabajo tiene como objetivo exponer y dar solución a un problema que los autores Ramadge y Wonham propusieron en el documento titulado *On the supremal controllable sublanguage of a given language* publicado en 1984 [12] donde proponen conceptos matemáticos del lenguaje de control y la existencia de controles de supervisión para el control de procesos que tienen eventos discretos. Para ejemplificar el funcionamiento del lenguaje de control, los autores proponen un problema sobre un gato y un ratón. El comportamiento de estos animales se expresa en forma de un sistema abstracto linear discreto y dinámico [13]. En términos simbólicos, tanto el gato como el ratón pueden representar robots móviles o dos vehículos de guiado automático (*Automatic Guidance Vehicle*) en sistemas flexibles de manufactura [14]. Ciertas condiciones se requieren cumplir para resolver el problema, incluyendo el uso de las redes de Petri y un software de simulación.

MATERIALES

Se tomó en consideración el problema de un gato y un ratón tomado de la página 1077 del artículo de Wonham y Ramadge [10]. Estos animalitos se colocan en un lugar que tiene cinco cuartos, de los cuales el gato inicialmente se encuentra en el cuarto número 2 y el ratón en el cuarto número 4. Cada puerta que conecta a todos los cuartos tiene especificada una dirección de movimiento exclusiva para el gato y otra para el ratón. Eso se ejemplifica gráficamente en la Figura 1, donde se observan los cuartos y las respectivas representaciones de movimiento del gato y el ratón. El símbolo c_n representa la transición del gato hacia los cuartos donde se puede mover y la letra m_n representa la transición del ratón a los cuartos donde se puede mover. Los cuartos 2 y 4 son incontrolables (sin restricciones) para el gato, ya que él es el único que puede entrar y salir de esos dos cuartos sin ningún problema. Sin embargo, en el problema se tienen que encontrar un control que cumpla con las siguientes condiciones:

1. El gato y el ratón nunca deben ocupar el mismo cuarto simultáneamente.
2. Que sea posible que el gato y el ratón regresen a su estado inicial, es decir, que el gato se encuentre en el cuarto 2 y el ratón en el cuarto 4.

Sin embargo, una tercera condición que los autores no tomaron en consideración, pero que Uzam [14] considera también que debe existir, es que el esquema del control debe permitir que el gato y el ratón se muevan lo más libremente posible con respecto a las restricciones de control, que en este caso, no existe para el gato en los cuartos 2 y 4 respectivamente. En otras palabras, el ratón no puede tampoco ocupar los cuartos 2 y 4 si el gato se encuentra en uno de estos cuartos.

 DOI: https://doi.org/10.18272/aci.v15i2.2932



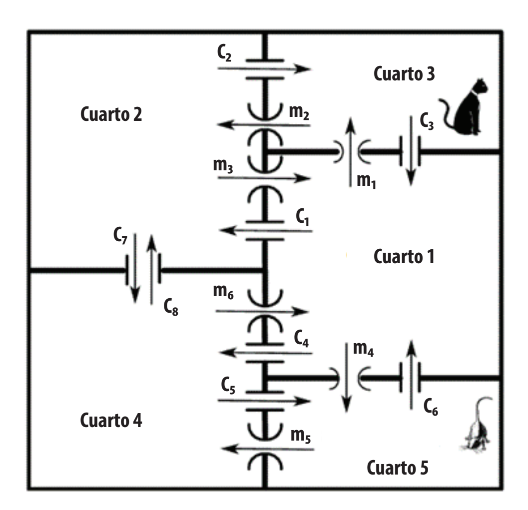


Figura 1. Representación del problema del gato y el ratón. Adaptado de Uzam [14]

Para resolver el problema se utilizó el programa HiPS el cual es una herramienta informática desarrollada por el Departamento de Ciencias Computacionales e Ingeniería de la Universidad Shinshu de Japón, escrito en lenguaje de programación C# y C++, que funciona en un ambiente de Windows. [15]. Como su nombre lo indica, es un simulador de redes de Petri. Se utilizó la versión 1.x del *software*. Además, para su operación se tomó en cuenta la información presentada en un video publicado en la plataforma YouTube [16].

MÉTODO

El primer paso fue simular mediante diseño el modelo de red de Petri de sistema de eventos discretos del problema. En la Figura 2 se representa el modelo de red de Petri hecho en el *software* HiPS para los movimientos del gato, donde se representa su estado inicial, las plazas de acuerdo con número de cuartos, el *token* inicial o marcaje (cuarto 3 donde se ubica inicialmente el gato), los arcos (conexiones de cada plaza) y las transiciones de cada cuarto. Las transiciones C_7 y C_8 son incontrolables, creando un movimiento libre del gato en los cuartos 2 y 4 respectivamente.

 DOI: https://doi.org/10.18272/aci.v15i2.2932



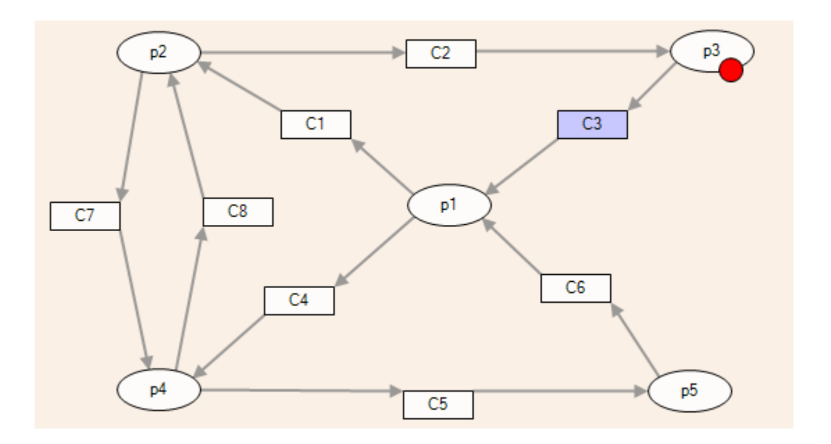


Figura 2. Representación de la red de Petri del estado inicial y del comportamiento posible del gato hecho en software HiPS

Por consiguiente, se procedió a simular en el *software* de HiPS de la misma manera el sistema inicial de la red de Petri de los movimientos del ratón, con el token en la plaza 5 inicial, las plazas, los arcos y las transiciones como lo muestra la Figura 3.

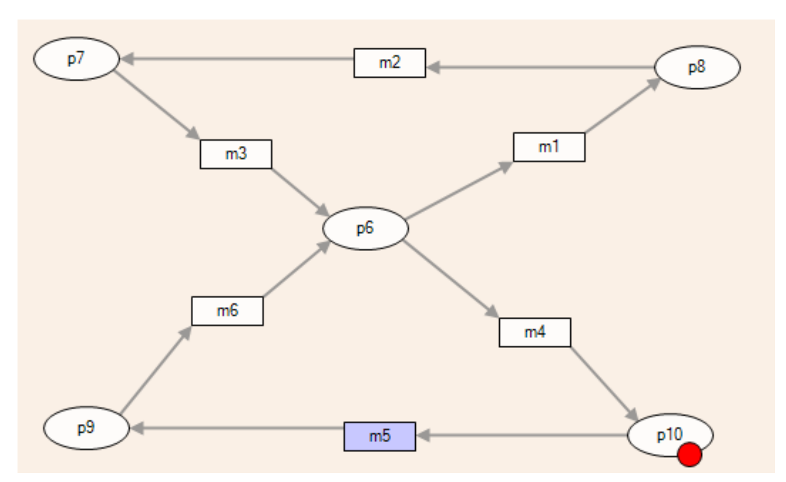


Figura 3. Representación de la red de Petri del estado inicial y del comportamiento posible del ratón hecho en software HiPS

Después se procedió a definir los estados que no pueden presentarse con base en las condiciones que se presentan en el problema, que son tres especificaciones. La primera especificación es que el gato y el ratón no deben de estar en un mismo cuarto al mismo tiempo, lo cual indica que no pueden estar los marcadores en el mismo lugar al mismo tiempo (porque el gato se come al ratón). También, se debe de considerar las transiciones incontrolables de C_7 y C_8 debido a que el gato esta en los cuartos 2 y 4 de manera libre, así que el ratón no puede estar en el cuarto 2 si el gato está en el cuarto 4 y viceversa.

 DOI: https://doi.org/10.18272/aci.v15i2.2932



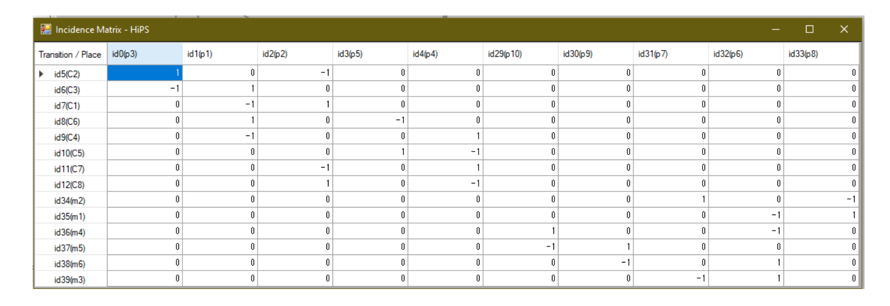
Como siguiente paso se procedió a realizar una tabla de alcanzabilidad y covertura de las dos redes de Petri en el *software* HiPS, tal como lo muestra la Figura 4. Dicha tabla es un modelo matemático para describir los marcadores asignados a cada plaza que pueden ser posiblemente en las plazas de toda la red [5].

```
number->Marking
0->(1, 0, 0, 0, 0, 1, 0, 0, 0, 0)
1->(0, 1, 0, 0, 0, 1, 0, 0, 0, 0)
2->(0, 0, 1, 0, 0, 1, 0, 0, 0, 0)
3->(0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
4->(0, 0, 0, 1, 0, 1, 0, 0, 0, 0)
5->(0, 0, 0, 1, 0, 0, 1, 0, 0, 0)
6->(0, 1, 0, 0, 0, 0, 1, 0, 0, 0)
7->(0, 0, 1, 0, 0, 0, 1, 0, 0, 0)
8->(1, 0, 0, 0, 0, 0, 1, 0, 0, 0)
9->(1, 0, 0, 0, 0, 0, 0, 0, 1, 0)
10->(0, 1, 0, 0, 0, 0, 0, 0, 1, 0)
11->(0, 0, 1, 0, 0, 0, 0, 0, 1, 0)
12->(0, 0, 0, 0, 1, 0, 0, 0, 1, 0)
13->(0, 0, 0, 1, 0, 0, 0, 0, 1, 0)
14->(0, 0, 0, 1, 0, 0, 0, 0, 0, 1)
15->(0, 1, 0, 0, 0, 0, 0, 0, 0, 1)
16->(0, 0, 1, 0, 0, 0, 0, 0, 0, 1)
17->(1, 0, 0, 0, 0, 0, 0, 0, 0, 1)
18->(1, 0, 0, 0, 0, 0, 0, 1, 0, 0)
19->(0, 1, 0, 0, 0, 0, 0, 1, 0, 0)
20->(0, 0, 1, 0, 0, 0, 0, 1, 0, 0)
21->(0, 0, 0, 0, 1, 0, 0, 1, 0, 0)
22->(0, 0, 0, 1, 0, 0, 0, 1, 0, 0)
23->(0, 0, 0, 0, 1, 0, 0, 0, 0, 1)
24->(0, 0, 0, 0, 1, 0, 1, 0, 0, 0)

placeID: (id0,id1,id2,id3,id4,id29,id30,id31,id32,id33)
placeNAME: (p3,p1,p2,p5,p4,p10,p9,p7,p6,p8)
```

Figura 4. Resultados de la alcanzabilidad y la covertura del modelo de la red de Petri del gato y el ratón en HiPS

A continuación, se realizó la matriz de incidencia de los dos modelos de las redes de Petri (gato y ratón), según se representa en la Figura 5 [4].



Incidence Matrix - HiPS

Transition / Place	id0(p3)	id1(p1)	id2(p2)	id3(p5)	id4(p4)	id29(p10)	id30(p9)	id31(p7)	id32(p6)	id33(p8)
id5(C2)	1	0	−1	0	0	0	0	0	0	0
id6(C3)	−1	1	0	0	0	0	0	0	0	0
id7(C1)	0	−1	1	0	0	0	0	0	0	0
id8(C6)	0	1	0	−1	0	0	0	0	0	0
id9(C4)	0	−1	0	0	1	0	0	0	0	0
id10(C5)	0	0	0	1	−1	0	0	0	0	0
id11(C7)	0	0	−1	0	1	0	0	0	0	0
id12(C8)	0	0	1	0	−1	0	0	0	0	0
id34(m2)	0	0	0	0	0	0	0	1	0	−1
id35(m1)	0	0	0	0	0	0	0	0	−1	1
id36(m4)	0	0	0	0	0	1	0	0	−1	0
id37(m5)	0	0	0	0	0	−1	1	0	0	0
id38(m6)	0	0	0	0	0	0	−1	0	1	0
id39(m3)	0	0	0	0	0	0	0	−1	1	0

Figura 5. Matriz de incidencia de los dos modelos de las redes de Petri (gato y ratón)

Se identificaron los estados donde no causaba ningún conflicto con las condiciones que se registraron en el problema, donde el gato y el ratón no ocupaban el mismo cuarto, ni que el ratón estuviera en los cuartos 2 y 4 cuando el gato estuviera en cualquiera de estos dos cuartos, los cuales se presentan en la Tabla 1.

DOI: https://doi.org/10.18272/aci.v15i2.2932



Tabla 1. Lista de los estados sin restricciones del problema del gato y el ratón, representando sus marcas y plazas para que se cumplan las condiciones del problema planteado

0->(1, 0, 0, 0, 0, 1, 0, 0, 0, 0)
1->(0, 1, 0, 0, 0, 1, 0, 0, 0, 0)
2->(0, 0, 1, 0, 0, 1, 0, 0, 0, 0)
3->(0, 0, 0, 0, 1, 1, 0, 0, 0, 0)
4->(1, 0, 0, 0, 0, 0, 1, 0, 0, 0)
5->(1, 0, 0, 0, 0, 0, 0, 0, 1, 0)

Posteriormente se procedió a construir otro modelo de Petri en HiPS para controlar las transiciones de acuerdo con los estados que no debería tener restricciones del problema. Es decir, se corrió el modelo a partir de la eliminación de las restricciones y las transiciones se ejecutaron sin bloqueos. El modelo se refleja en la Figura 6.

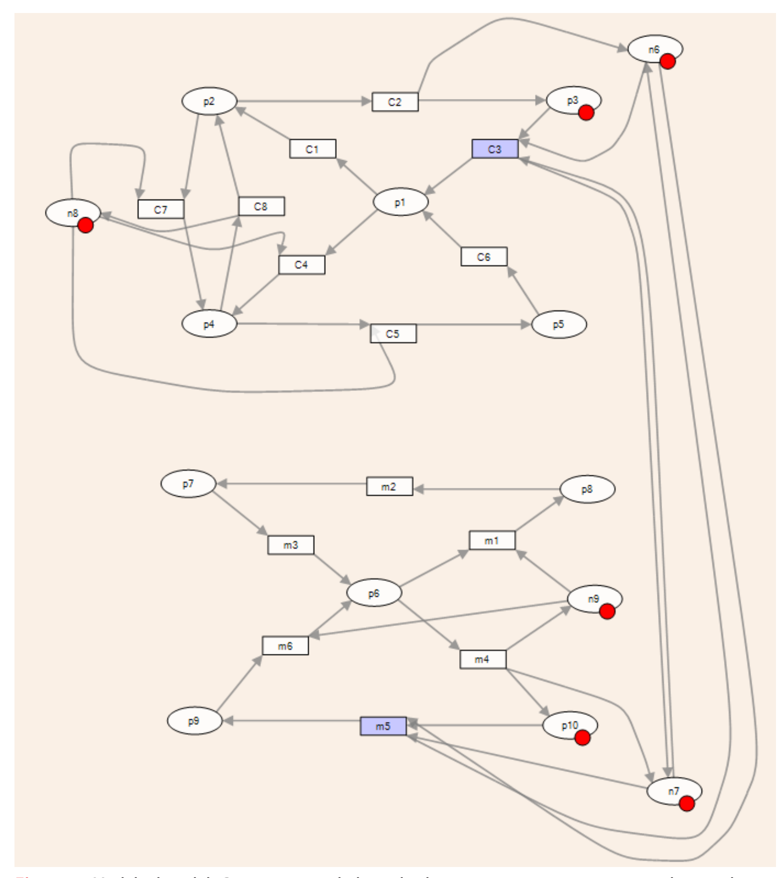


Figura 6. Modelo de red de Petri con controladores de algunas transiciones para ejecutar los estados sin restricciones

 DOI: https://doi.org/10.18272/aci.v15i2.2932



RESULTADOS

Al correr el programa HiPS con el modelo planteado, se obtuvo la matriz de incidencia del modelo representado en la Figura 7.

Incidence Matrix - HiPS

Transition / Place	id0(p3)	id1(p1)	id2(p2)	id3(p5)	id4(p4)	id29(p10)	id30(p9)	id31(p7)	id32(p6)	id33(p8)	id52(n6)	id53(n8)	id54(n7)	id55(n9)
id5(C2)	1	0	−1	0	0	0	0	0	0	0	1	0	0	0
id6(C3)	−1	1	0	0	0	0	0	0	0	0	−1	0	0	0
id7(C1)	0	−1	1	0	0	0	0	0	0	0	0	0	0	0
id8(C6)	0	1	0	−1	0	0	0	0	0	0	0	0	0	0
id9(C4)	0	−1	0	0	1	0	0	0	0	0	0	−1	0	0
id10(C5)	0	0	0	1	−1	0	0	0	0	0	0	−1	0	0
id11(C7)	0	0	−1	0	1	0	0	0	0	0	0	−1	0	0
id12(C8)	0	0	1	0	−1	0	0	0	0	0	0	1	0	0
id34(m2)	0	0	0	0	0	0	0	1	0	−1	0	0	0	0
id35(m1)	0	0	0	0	0	0	0	0	−1	1	0	0	0	−1
id36(m4)	0	0	0	0	0	1	0	0	−1	0	0	0	1	1
id37(m5)	0	0	0	0	0	−1	1	0	0	0	0	0	−1	0
id38(m6)	0	0	0	0	0	0	−1	0	1	0	0	0	0	−1
id39(m3)	0	0	0	0	0	0	0	−1	1	0	0	0	0	0

Figura 7. Matriz de incidencia del nuevo modelo con los controladores

Al ejecutar el programa HiPS para analizar el alcance y cobertura del modelo de Petri propuesto, arrojó los mismos estados sin restricciones que se identificaron anteriormente, según lo representa la Figura 8.

```
number->Marking
0->(1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 1, 1, 1)
1->(0, 1, 0, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1)
2->(0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 1, 1)
3->(0, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 1, 1)
4->(1, 0, 0, 0, 0, 0, 1, 0, 0, 0, 1, 1, 0, 1)
5->(1, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0)
```

Figura 8. Tabla de alcanzabilidad y cobertura del nuevo modelo de Petri según el análiss de HiPS

También, con respecto al bloqueo o punto muerto de la simulación del sistema mediante la dinámica y el movimiento de los marcadores en cada plaza, se analizó esta propiedad en el software HiPS y el resultado fue que no encontró ningún bloqueo en la simulación, según lo muestra la Figura 9.

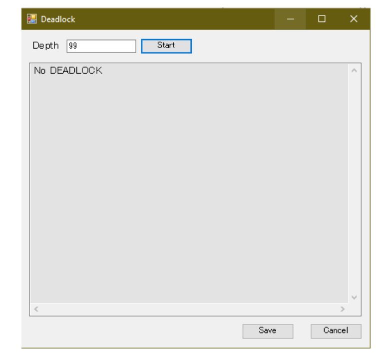


Figura 9. HiPS no encontró ningún bloqueo o punto muerto en la simulación del modelo de Petri

DOI: https://doi.org/10.18272/aci.v15i2.2932



Por último, para comprobar la reversibilidad de los estados, es decir, que el gato y el ratón volvían a su estado inicial, en el software de HiPS se hizo el análisis de esta condición, arrojando lo resultados en la Figura 10.

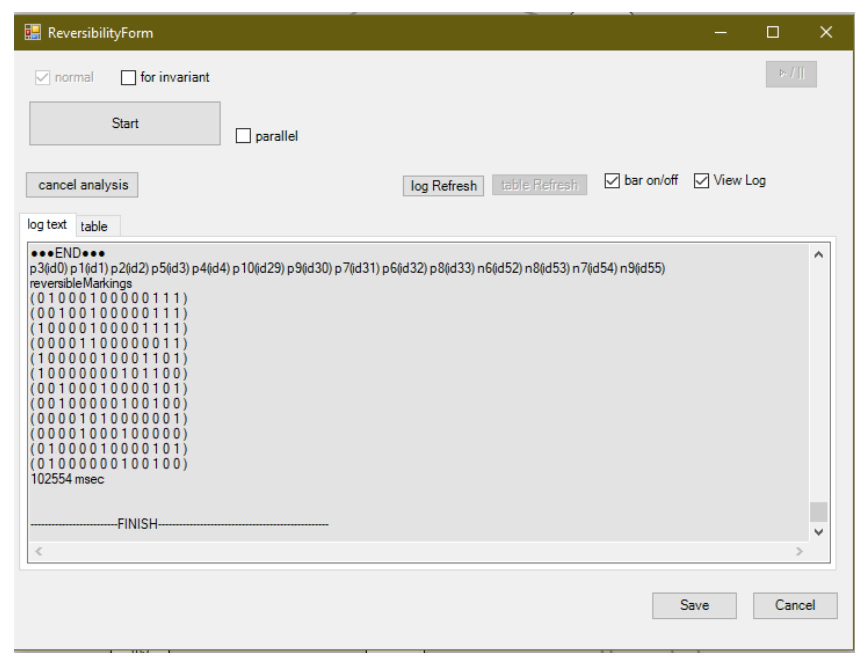


Figura 10. Análisis de reversibilidad de las marcas

Por lo tanto, se concluye con la información presentada que con el uso del *software* HiPS se pudo realizar con éxito el resultado del problema planteado por Wonham y Ramadge. Para el lector interesado, los archivos de los datos y métodos utilizados se ponen a disposición en el siguiente link de Github: https://github.com/saulmanuel/HiPS-Gato-y-Raton.

CONCLUSIONES

Las redes de Petri pueden ser una herramienta práctica de modelación para varios tipos de sistemas dinámicos, incluidos aquellos donde se manifiestan eventos discretos. Desde sus orígenes, se han publicado varios artículos sobre sus usos y aplicaciones [5].

El problema expuesto en este artículo es solamente un ejemplo de la aplicación práctica de las redes de Petri tomando la ventaja de las tecnologías de información que permiten la modelación de una manera rápida y automatizada. Se enfatiza la importancia de incrementar el estudio de la aplicación de las redes de Petri en otras áreas del conocimiento, donde los sistemas dinámicos y eventos discretos puedan ser modelados. También, el problema puede servir de referente para desarrollar herramientas útiles para el análisis de modelos donde se requiere tomar en consideración restricciones o estados conflictivos de un sistema para encontrar la solución óptima que permita la "ruta de viaje" adecuado de los marcadores.

DOI: https://doi.org/10.18272/aci.v15i2.2932





Además, el éxito del uso de sistemas informáticos de simulación donde se combinan métodos de modelación de sistemas dinámicos depende de la habilidad del usuario de aplicar el modelo en las condiciones indicadas, los niveles de interpretación del mismo y la disponibilidad de la tecnología.

CONFLICTO DE INTERÉS

La publicación de este artículo científico se ha llevado a cabo con total transparencia y rigor académico, garantizando la ausencia de conflictos de interés que puedan influir en los resultados o conclusiones presentados. El autor ha seguido rigurosamente los principios éticos y las normas de buena práctica científica, asegurando que no exista ningún vínculo financiero, personal o profesional que pueda sesgar la objetividad del estudio. De esta manera, se ha preservado la integridad de la investigación y se ha promovido la confianza en los resultados presentados, brindando una contribución valiosa al conocimiento científico.



REFERENCIAS

[1] Kuntsevich, V. M., Gubarev, V. F., Kondratenko, Y. P., Lebedev, D. V. y Lysenko, V. P. (2018). *Control Systems: Theory and Applications*. River Publishers.

[2] Balemi, S. (1992). *Control of Discrete Event Systems: Theory and application*. ADAG Administration & Druck AG.

[3] Gunardi, Y. y Hanafi, D. (2022, julio 20-21). *Petri net modeling for mobile robot in sharp turning cases*. International Seminar of Intelligent Technology and its Applications (ISITIA). Surabaya, Indonesia. doi: https://doi.org/10.1109/ISITIA56226.2022.9855287

[4] Murata, T. (1989). Petri nets: Properties, analysis and applications. *Proceedings of the IEEE, 77*(4), 541-580. doi: https://doi.org/10.1109/5.24143

[5] Wang, J. (2007). Petri Nets for Dynamic Event-Driven System Modeling. En P. A. Fishwick (Ed.), *Handbook of Dynamic System Modeling*. Monmouth University. doi: https://doi.org/10.1201/9781420010855

[6] Zurawski, R. y Zhou, M. (1994). Petri nets and industrial applications: A tutorial. *IEEE Transactions on Industrial Electronics, 41*(6), 567-583. doi: https://doi.org/10.1109/41.334574

[7] Restrepo, P. L. A. (2011). Un método computacional para la detección y caracterización de conflictos en redes de Petri. *Revista Ingenierías Universidad de Medellín, 10*(19), 189-199. https://repository.udem.edu.co/handle/11407/931

[8] Ramadge, P. J. (1983). *Control and Supervision of Discrete Event Processes*. Department of Electrical Engineering, University of Toronto.

[9] Ramadge, P. J. y Wonham, W. M. (1982, diciembre 8-10). *Supervision of discrete event processes*. 21st IEEE Conference on Decision and Control. Orlando, FL, EE.UU. doi: https://doi.org/10.1109/CDC.1982.268351

[10] Ramadge, P. J. y Wonham, W. M. (1987). Supervisory control of a class of discrete event processes. *SIAM Journal on Control and Optimization, 25*(1), 206-230. doi: https://doi.org/10.1137/0325013

[11] Wonham, W. M. y Ramadge, P. J. (1987). On the Supremal Controllable Sublanguage of a given Language. *SIAM Journal on Control and Optimization, 25*(3), 637-659. doi: https://doi.org/10.1137/0325036

[12] Wonham, W. M. y Ramadge, P. J. (1984, diciembre 12-14). *On the supremal controllable sublanguage of a given language*. 23rd Conference on Decision and Control. Las Vegas, NV, EE.UU. doi: https://doi.org/10.1109/CDC.1984.272178

[13] Čapkovič, F. (1993). A Petri Nets-Based Approach to the Maze Problem Solving. En S. Balemi, P. Kozák y S. Smendiga (Eds.), *Discrete Event Systems: Modeling and Control* (pp. 173-179). Birkhäuser Basel. doi: https://doi.org/10.1007/978-3-0348-9120-2_15

[14] Uzam, M. (2004). Synthesis of feedback control elements for discrete event systems using Petri net models and theory of regions. *International Journal of Advance Technology, 24*, 48-69. doi: https://doi.org/10.1007/s00170-003-1715-x

[15] Harie, Y., Mitsui, Y., Fujimori, K., Batajoo, A. y Wasaki, K. (2017, octubre 24-27). *HiPS: Hierarchical Petri Net design, simulation, verification and model checking tool*. IEEE 6th Global Conference on Consumer Electronics (GCCE). Nagoya, Japón. doi: https://doi.org/10.1109/GCCE.2017.8229199

[16] KWasaki1967. (2017, junio 4). *HiPS: Hierarchical Petri net Simulator* [video]. YouTube. https://www.youtube.com/watch?v=usF1JrQegOE

 DOI: https://doi.org/10.18272/aci.v15i2.2932